Iranian Society of Acoustics and Vibration
The 13th ISAV2023 International Conference on Acoustics and Vibration
20 , 21 Dec 2023 Tehran - Iran

Leveraging Koopman Operator and Deep Neural Networks for Parameter Estimation and Future Prediction of Duffing Oscillators

Yassin Riyazi [a],NavidReza Ghanbari [a], Arash Bahrami [a*]

[a] *M.Sc. Student, School of Mechanical Engineering, College of Engineering, University of Tehran, Tehran, Iran.*

[b] *Assistant Professor, School of Mechanical Engineering, College of Engineering, University of Tehran, Tehran, Iran.*

* *Corresponding author e-mail: arash.bahrami@ut.ac.ir (A. Bahrami)*

Abstract

The study of nonlinear dynamical systems has been a cornerstone in various scientific and engineering fields due to their widespread applications in modeling real-world phenomena. Traditional methods for analyzing and predicting the behavior of such systems often involve complex mathematical techniques and numerical simulations. This paper introduces a novel approach that combines the power of the Koopman Operator and deep neural networks to generate a linear representation of the Duffing oscillator, enabling effective parameter estimation and accurate prediction of its future behavior. Furthermore, a modified loss function is proposed to enhance the training process of the deep neural network. The synergy of the Koopman Operator and deep neural networks not only simplifies the analysis of nonlinear systems but also opens a promising avenue for advancing predictive modeling in various fields.

Keywords: Koopman Operator; Parameter Estimation; Duffing oscillator; deep neural networks; nonlinear dynamical systems; predictive modeling.

1. Introduction

Nonlinear dynamical systems, recognized for their intricate and sometime chaotic behavior, permeate the realms of natural phenomena and technological applications. They transcend the simplicity of linear systems, giving rise to phenomena such as bifurcations, limit cycles, and chaotic attractors. These systems have long captivated the interest of scientists and engineers, presenting substantial challenges in understanding, characterizing, and predicting their trajectories. Across diverse fields, from physics and biology to economics and engineering, nonlinear systems underscore the fundamental complexity of our world.

At the heart of this intricate landscape lies the Duffing Oscillator [1], an iconic archetype of nonlinear dynamical systems. Its versatility enables it to emulate a wide spectrum of behaviors, making it a pertinent model for various physical phenomena. From capturing the subtle interplay of mechanical vibrations in structures subjected to external forces to mirroring the rhythmic patterns of biological oscillations, the Duffing oscillator encapsulates the essence of nonlinear dynamics.

Traditionally, dissecting and forecasting the behavior of Duffing oscillators have relied on a combination of analytical techniques and numerical simulations. While these methods provide valuable insights, they often encounter limitations in handling nonlinear intricacies with precision. Analytical solutions may prove elusive or algebraically expensive, especially for higher-dimensional or strongly nonlinear systems. On the contrary, numerical simulations, though powerful, demand extensive computational resources and face challenges in long-term predictions due to inherent numerical errors and uncertainties.

The Koopman Operator [2], with its inherent structure involving a mapping to a higher dimension, a linear transformation, and an inverse mapping, bears resemblance to the structure of autoencoders [3]. Acknowledging the pioneering work of S. L. Brunton and J. N. Kutz, who introduced the SINDY method [4], we appreciate the potential for leveraging the generality offered by deep learning to successfully identify systems [5], [6].

While the sole use of a neural network proves accurate and meets our requirements, it does not guarantee the exclusive confinement of the Koopman Operator to a designated linear layer. Consequently, the entire network structure incorporates elements of mapping, linear transformation, and inverse mapping simultaneously. This challenges the utility of using the linear layer weights as a representation of the system, as they only encapsulate a portion of the Koopman Operator.

To overcome these challenges, we introduce an innovative approach that capitalizes on the synergy between the Koopman Operator and Deep Neural Networks [7]–[9]. This groundbreaking fusion is aimed at converting the Duffing oscillator into a linearized representation, offering promising solutions to the intricacies encountered in traditional methods. By harnessing the computational power of deep learning and the Koopman operator's capability to provide a linear representation of nonlinear systems [10], our approach enables a more accurate Koopman linearized representation of system behavior.

Our approach not only streamlines the analysis of nonlinear systems but also extends its applicability across a diverse spectrum of domains. It ushers in a new era in predictive modeling by opening doors to effective parameter estimation and precise future predictions, addressing the challenges posed by the inherent complexity of nonlinear systems.

In the following sections, we dig into the foundational principles of Koopman Operator theory and the adaptability of deep neural networks. We show how their fusion forms a compelling framework for analyzing and predicting the behavior of Duffing oscillators. We outline the process of transforming Duffing oscillator dynamics into a linear representation and introduce a modified loss function designed to enhance the generality of the Koopman linear representation of the dynamical system within this context. Through numerical validation and comparisons with traditional methods, we demonstrate the efficacy of our approach in providing accurate predictions for the future behavior of Duffing oscillators. Ultimately, this work enriches our understanding of nonlinear dynamics and offers a powerful tool with transformative potential across scientific, engineering, and practical applications.

2. Koopman Operator and Its Application

The exploration of dynamical systems has long been a cornerstone in understanding complex behaviors in various scientific and engineering disciplines. Traditionally, the analysis of these systems has been deeply entwined with the concept of state space, where the evolution of a system is represented by trajectories in the space of its state variables. However, the inherent nonlinearity of many real-world systems often makes their analysis and prediction challenging using conventional techniques.

In recent years, the Koopman Operator has emerged as a powerful mathematical tool that provides a fresh vantage point for studying dynamical systems. Rooted in functional analysis, the Koopman Operator introduces a paradigm shift by transitioning the focus from the state space to the space of observable functions. Doing so allows us to view the system's evolution in a linear framework, even when dealing with inherently nonlinear systems. This perspective offers a new lens through which we can gain insights into the dynamics of complex systems.

2.1 Dynamical System Representation

Consider a dynamical system described by a set of state variables $\mathbf{x}(t)$, which evolve over time t. Mathematically, we can represent this as:

$$\frac{d\mathbf{x}}{dt} = f\left(\mathbf{x}, t\right) \tag{1}$$

where

- n is the state vector dimension.
- $\mathbf{x} \in \mathbb{R}^n$ is the state vector representing the system's state variables.
- $t \in \mathbb{R}^+$ is time.
- $f\left(\mathbf{x}\right): \mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued function describing how the state variables change over time.

2.2 Koopman Operator Transformation

The Koopman operator, denoted as $\mathcal{K}$, is an infinite-dimensional linear operator that acts on observables or functions of the state variables. Let $g\left(x\right)$ be such an observable. The Koopman Operator maps this observable from the state space to a higher-dimensional space:

$$\mathcal{K}g\left(\mathbf{x}\right) = g\left(f\left(\mathbf{x}\right)\right) \tag{2}$$

Where:

- m supposed to be infinite-dimensional but in numerical approximation a value will be assigned.
- $\mathcal{K}(.): \mathbb{C}^m \to \mathbb{C}^m$ is the Koopman Operator Generator.
- $g\left(\mathbf{x}\right): \mathbb{R}^n \to \mathbb{C}^m$ is an observable or function defined on the state space.
- $g\left(f\left(\mathbf{x}\right)\right)$ represents the observable after the system evolves according to $f\left(\mathbf{x}\right)$.

2.3 Koopman Operator in Discrete Time

In discrete-time dynamical systems, the Koopman Operator is applied at discrete time steps. For these systems Equation (2) may be represented as:

$$\mathcal{K}g\left(\mathbf{x}_k\right) = g\left(\mathbf{x}_{k+1}\right) \tag{3}$$

where:

- $\mathbf{x}_k$ represents the state of the system at time k.
- $\mathbf{x}_{k+1}$ represents the state of the system at the next time step $k+1$.

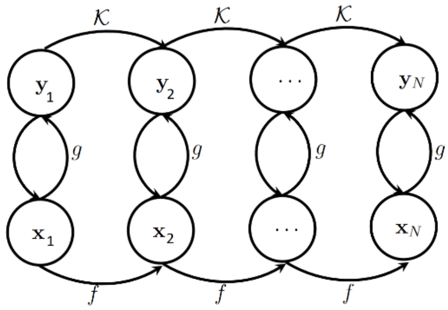

Figure 1. Koopman Operator Evolution and a discrete dynamical system

In Figure 1. an important relationship becomes apparent: the transformation of data x_t from its low-dimensional representation in Euclidean space to an infinite-dimensional Hilbert space is facilitated through the utilization of Koopman observables $g(x_t)$. Leveraging this Koopman transformation allows for the mapping of y_t to y_{t+1} via a linear matrix transformation. Furthermore, employing the inverse Koopman observable mapping $g^{-1}(y_{t+1})$ enables the derivation of x_{t+1}. With help of Koopman operator $f(x_t) = g^{-1}(\mathcal{K} \times g(x_t)) = x_{t+1}$.

3. Coupling Koopman Operators with Deep Neural Networks

Deep neural networks have showcased remarkable abilities in approximating intricate functions and mastering complex patterns from data. One of the key challenges encountered in the realm of Koopman Operators is the identification of suitable observable functions. In methods such as DMD [11], the observable function is typically the identity function, and in extended DMD (EDMD), observable functions take the form of polynomials or trigonometric functions. While these approaches are straightforward and accurate, they exhibit resilience to noise and initial conditions.

In this study, we combine a deep neural network with the Koopman operator, thereby generating a linearized representation of the Duffing oscillator. This neural network effectively learns the intricate relationship between system parameters and observed behaviors, facilitating efficient parameter estimation. Moreover, the neural network undergoes training to predict the future trajectory of the Duffing oscillator, thereby equipping us with a valuable tool for forecasting system behavior.

neural networks are not necessarily always better than feature crosses, but neural networks do offer a flexible alternative that works well in many cases.

3.1 Data acquisition

The Duffing oscillator is a dynamical system described by the following second-order differential equation:

$$\frac{d^2x}{dt^2} + \delta \frac{dx}{dt} + \alpha x + \beta x^3 = \gamma \cos(\omega t) \tag{5}$$

Here:

- x represents the displacement of the oscillator from its equilibrium position.

- δ denotes the damping coefficient.
- α is the linear stiffness coefficient.
- β characterizes the nonlinearity in the system.
- γ is the amplitude of the external driving force.
- ω is the angular frequency of the driving force.

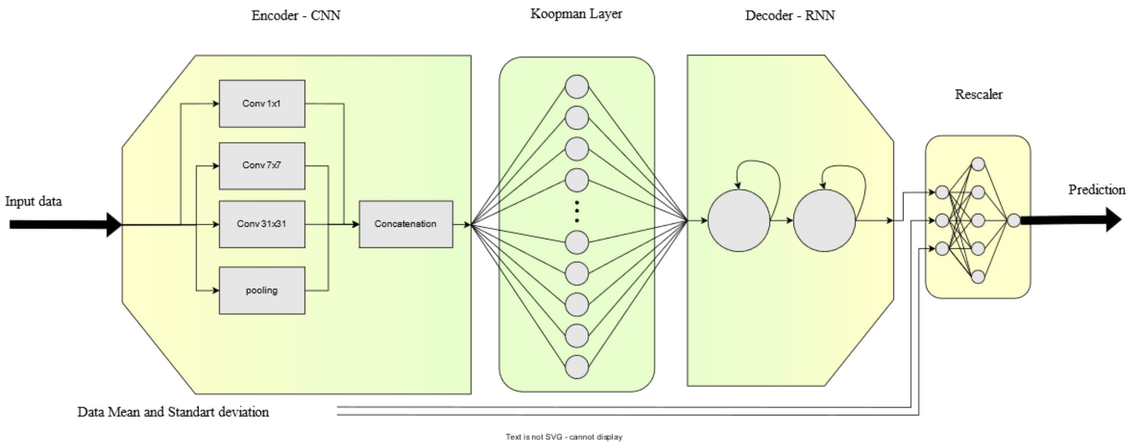

Figure 2. Neural Network diagram

Duffing oscillator solution has been generated using Runge-Kuta method [12] and initial condition for Solving the equation is $x_0 = 1.5$ and $v_0 = -1.5$.

A normal distributes with noise in range of [-0.5,0.5] added to data to simulate real world data.

The elucidation of the numerical solution has been provided within the Appendix dedicated to Numerical simulation.

3.2 Data Normalization

Given the nature of regression, it is advisable to normalize the data before training the neural network. This normalization is crucial because even small variations in the input data can lead to significant changes in the output, potentially reducing the model's robustness against changes in input conditions. In our approach, we employ data normalization prior to feeding it into the convolutional neural network (CNN) architecture. During this process, we pass the statistical properties of the data, such as its mean and variance, through the network.

In Figure 2. Neural Network diagram, after the recurrent neural network (RNN) block, the data is remapped to its original statistical properties before being passed to the Rescaler Block. The purpose of the Rescaler Block is to further reduce the variation in the output, ultimately leading to a more stable and controlled model response.

3.3 Structure

The architecture of the model as shown is Figure 2. Neural Network diagram is structured around an Encoder-Decoder paradigm, which effectively captures the essence of complex dynamics. Specifically, the Encoder component is meticulously designed, featuring a sequence of Inception Blocks [13] in a convolutional neural network (CNN) [14]. These Inception Blocks serve as robust feature extractors, enabling the model to discern intricate patterns and relevant features from the input data.

Following the Inception-based Encoder, a pivotal transformation takes place through a linear layer. This linear layer assumes a distinct role within the architecture, embodying the essence of the Koopman Operator evolution function $\mathcal{K}$. It is important to note that this linear layer operates without an activation function and bias, preserving the linear nature of the Koopman operator's transformation.

Transitioning from the Koopman Operator layer, the architecture takes an intriguing turn with the integration of a two-layer Long Short-Term Memory (LSTM) [15] network. This LSTM component acts as the Decoder, expertly leveraging its sequential memory to unravel the transformed linearized representation. This sequence-to-sequence modeling approach facilitates the reconstruction of the system's temporal evolution, a crucial aspect in capturing its intricate behaviors.

Table 1. CNN parameters. Out hyper parameter is 20

	Sequential Blocks	In Channels	Out Channels	kernel size	padding
branch1x1	Conv 1×1	1	out	1	0
	ReLU	-	-	-	-
branch7x7	Conv 1×1	1	out	1	0
	ReLU	-	-	-	-
	Conv 7×7	out	out	7	3
	ReLU	-	-	-	-
branch31x31	Conv 1×1	1	out	1	0
	ReLU	-	-	-	-
	Conv 31×31	out	out		15
	ReLU	-	-	-	-
branch pool	MaxPool1d	1	1	3	1
	-	-	-	-	-
	Conv 1×1	1	out	1	0
	ReLU	-	-	-	-

3.4 Training

The network is trained end-to-end, without the need for custom loss functions or specialized training algorithms. However, it is important to note that the evolution function of the Koopman Operator does not remain confined solely to the Koopman part; instead, it spreads throughout the network. In a sense, the network operates as a black box, handling this evolution internally.

To address this issue and restrict the Koopman Operator's influence exclusively to the Koopman linear layer, a two-stage training algorithm has been proposed. In this algorithm, after each optimization step:

1. The weights of all layers except the Koopman Linear Layer are frozen.
2. The output of the Koopman Linear Layer is calculated for time steps n_0 to n_{KPH} (KPH is the a Hyper parameters and due to cost of calculating matrix power 20 was selected).
3. The weights of the Koopman Linear Layer are updated based on the linearity property. This update aims to minimize the prediction error of the nth output .

$$\sum_{n}^{KPH} \mathcal{L}\left(g\left(x_{n_0}\right) \times \left(W_{Koopman}^{n}\right)^{T}, g\left(x_{n}\right) \times \left(W_{Koopman}\right)^{T} \right) \tag{4}$$

By implementing this two-stage training process, we ensure that the Koopman Operator's influence is confined and utilized specifically within the Koopman linear layer, enhancing the network's predictive accuracy and control.

Table 2. Optimizer and loss parameters

		Stage 1	Stage 2
Optimizer	Type	SGD	SGD
	Learning Rate	5.00E-02	5.00E-04
	momentum	0.9	-
	weight decay	1.00E-04	-

loss	Type	MSE	MSE

4. Results and Discussion

Numerical results demonstrate the effectiveness of the proposed approach. The combination of Koopman operator-based linearization and deep neural networks yields impressive results in terms of parameter estimation accuracy and future prediction. During the Networks training input horizon for extracting feature was 200 previous sample.

4.1 Simple periodic $\gamma = 0.2$

For more information visit Appendix, A and B.

Neural network is robust against noise with range (-0.5,0.5) to (-0.07,0.07)

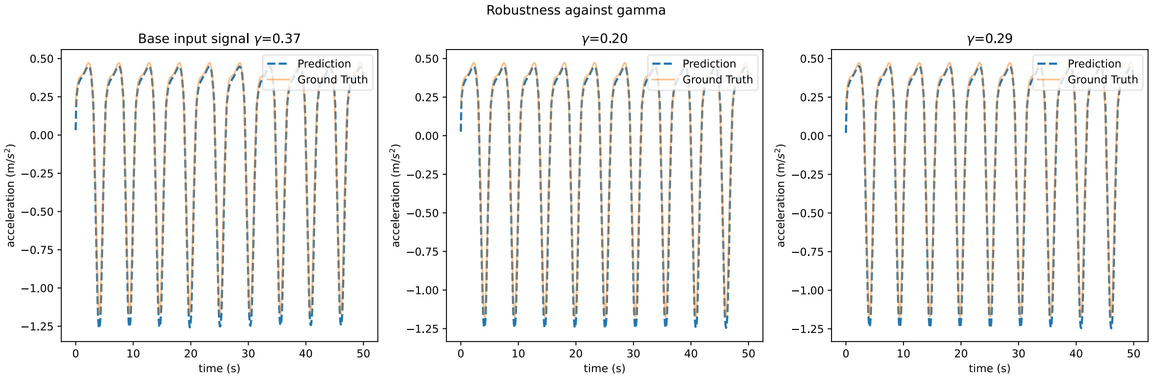

Figure 3. robust against Initioal Condition Variation

When considering various initial conditions while keeping the parameters (gammas) constant, the neural networks perform adequately well. However, it is important to note that ensuring the networks do not achieve a loss lower than 0.01 is advisable, as excessively low losses may increase the risk of overfitting.

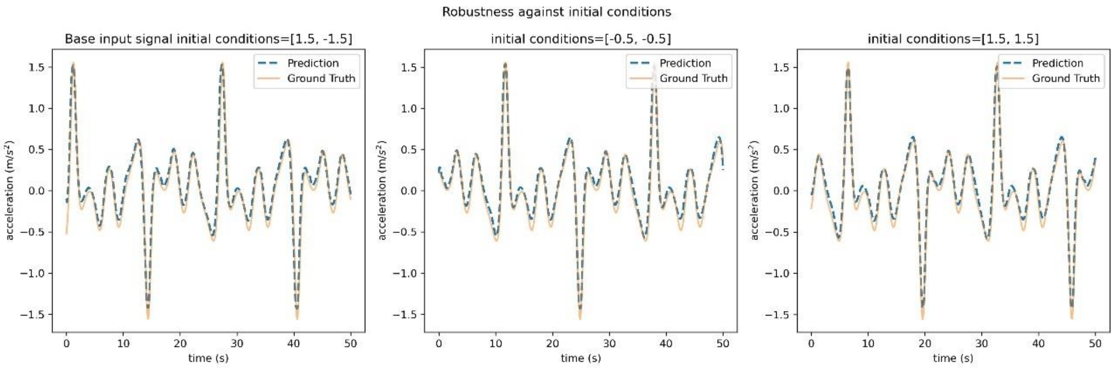

Figure 4. robust against initial condition

Different values of gammas do not yield satisfactory results, as datasets corresponding to distinct gammas exhibit entirely different structures. Introducing diverse gamma values during training may also lead to suboptimal outcomes. An alternative approach worth exploring is the expansion of the network's capacity, which could potentially enhance its performance under such circumstances.

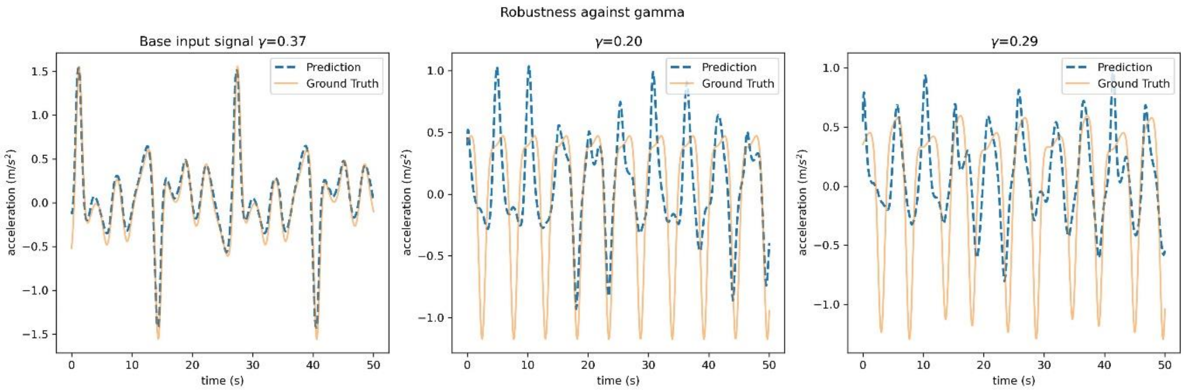

Figure 5. robust against *γ*

Various sampling rates yielded suboptimal results as well.

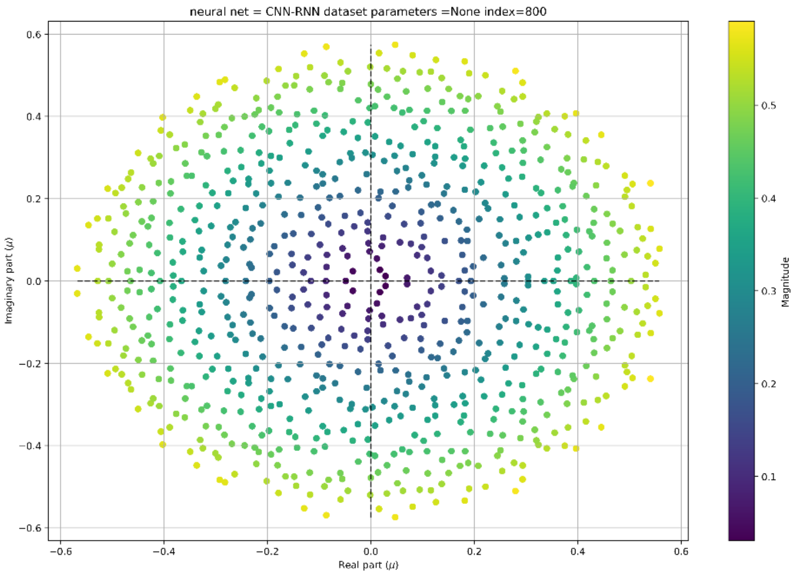

Figure 6. Koopman Layer Eigen values

5. Conclusion

In a broader context, the combination of the Koopman Operator and Neural Networks shows significant potential. The network has effectively captured the underlying data structure, and with further refinement, it has the capacity to generalize effectively to more complex problem domains.

6. Future studies

In the future research endeavors, several promising avenues can be explored:

1. Transitioning from Offline to Online Processing: An interesting prospect is the shift from offline to online data processing. Implementing real-time data analysis and prediction systems can enhance the applicability and timeliness of the models developed.
2. Integration of Gradient Clipping: The inclusion of gradient clipping techniques in training neural networks should be considered. This can help mitigate issues related to exploding gradients and improve model stability during training.
3. Incorporation of Mixture Density Networks for Confidence Estimation: Particularly in scenarios involving chaotic dynamics, the addition of Mixture Density Networks (MDNs) at the

model's output can provide valuable confidence estimates. This can enhance the reliability of predictions, especially when dealing with inherently uncertain or complex data.

4. Exploration of Small Transformer Models: The application of smaller-scale Transformer models warrants investigation. These models can serve a dual purpose: identifying any potentially missed data patterns within signals and subsequently utilizing these identified patterns for prediction tasks or noise reduction, thus improving overall data quality.

These avenues represent promising directions for advancing the research in this domain, with the potential to yield enhanced model performance and broader applications.

- Make sense of using Convolutions in Koopman Operator and prove that is a nonlinear function like Brunton's research
- Bring the Config of the Computer and the Flops required to train
- تبدیل معکوس تنها برای جواب های رو منیفلد کار میکند و وقتی نویز داریم دیگر تبدیل معکوس یکتا وجود ندارد و مهم نیست تابع واران چه باشد. ارجاع بده به شبکه ای که علی دایت ارایه داد

REFERENCES

[1] Hamel, "Georg Duffing, Ingenieur: Erzwungene Schwingungen bei veränderlicher Eigenfrequenz und ihre technische Bedeutung. Sammlung Vieweg. Heft 41/42, Braunschweig 1918. VI+134 S," *ZAMM - J. Appl. Math. Mech. / Zeitschrift für Angew. Math. und Mech.*, vol. 1, no. 1, 1921, doi: 10.1002/zamm.19210010109.

[2] B. O. Koopman, "Hamiltonian Systems and Transformation in Hilbert Space," *Proc. Natl. Acad. Sci.*, vol. 17, no. 5, 1931, doi: 10.1073/pnas.17.5.315.

[3] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science (80-.).*, vol. 313, no. 5786, 2006, doi: 10.1126/science.1127647.

[4] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 113, no. 15, 2016, doi: 10.1073/pnas.1517384113.

[5] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nat. Commun.*, vol. 9, no. 1, 2018, doi: 10.1038/s41467-018-07210-0.

[6] K. Champion, B. Lusch, J. Nathan Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 116, no. 45, 2019, doi: 10.1073/pnas.1906995116.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "2012 AlexNet," *Adv. Neural Inf. Process. Syst.*, 2012.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[9] M. Shafiq and Z. Gu, "Deep Residual Learning for Image Recognition: A Survey," *Applied Sciences (Switzerland)*, vol. 12, no. 18. 2022. doi: 10.3390/app12188972.

[10] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern Koopman Theory for Dynamical Systems," *SIAM Rev.*, vol. 64, no. 2, 2022, doi: 10.1137/21M1401243.

[11] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *J. Fluid Mech.*, vol. 656, 2010, doi: 10.1017/S0022112010001217.

[12] C. Runge, "Ueber die numerische Auflösung von Differentialgleichungen," *Math. Ann.*, vol. 46, no. 2, 1895, doi: 10.1007/BF01446807.

[13] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. doi: 10.1109/CVPR.2015.7298594.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, 1998, doi: 10.1109/5.726791.

[15] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, 1997, doi: 10.1162/neco.1997.9.8.1735.

Appendix

A) Numerical simulation

The Runge-Kutta method is a numerical technique used for solving ordinary differential equations (ODEs), such as those governing the behavior of the Duffing oscillator. It's commonly employed when analytical solutions are difficult to obtain. Here's an explanation of how the Runge-Kutta method can be applied to solve the Duffing oscillator equation in third person, including mathematical notation:

The Duffing oscillator is described by the second-order ordinary differential equation:

$$m\ddot{x}(t)+k\dot{x}(t)+cx(t)+\alpha x^{3}(t)=F(t) \tag{a}$$

Where:

- k is the stiffness constant.
- α is a coefficient that determines the strength of nonlinearity.
- $x(t)$ is the displacement of the oscillator at time t.
- c is the damping coefficient.
- $\dot{x}(t)$ represents the first derivative of $x(t)$ with respect to time$t$, which is velocity.
- m is the mass of the oscillator.
- $\ddot{x}(t)$ represents the second derivative of $x(t)$ with respect to time t, which is acceleration.
- $F(t)$ is the external force applied to the oscillator at time t.

To apply the Runge-Kutta method to solve this equation, by convert this second-order ODE into a system of first-order ODEs. and introducing a new variable, such as $v(t)$, to represent the velocity $\dot{x}(t)$. Then, two first-order ODEs become:

$$\begin{aligned}&\dot{x}(t)=v(t)\\&\dot{v}(t)=\frac{1}{m}\left[\mathrm{F}(t)-cv(t)-kx(t)-\alpha x^{3}(t)\right]\end{aligned} \tag{b}$$

To solve this equation using the Runge-Kutta method, the following steps are typically followed:

1. Discretization: Divide the time interval over which you want to solve the equation into small time steps. Let Δt represent the size of each time step, and create a time grid with time points $t_0, t_1, t_2, \ldots, t_n$ where $t_i = t_0 + i\Delta t_i$.
2. Initialization: Set the initial conditions for the displacement and velocity, x_0 and v_0, at t_0.
3. Iteration: For each time step i, perform the following calculations:
   3.1. Calculate the acceleration a_i at time t_i using the Duffing oscillator equation [reference to Duffing equation]
   3.2. Use the Runge-Kutta method to update the displacement and velocity for the next time step:

$$
\begin{aligned}
k_1 &= \Delta t\left(v_i\right)\\
l_1 &= \Delta t\left(a_i\right)\\
k_2 &= \Delta t\left(v_i+\frac{1}{2}l_1\right)\\
l_2 &= \frac{\Delta t}{m}\left(F\left(t_i+\frac{\Delta t}{2}\right)-c\left(v_i+\frac{l_1}{2}\right)-k\left(x_i+\frac{k_1}{2}\right)-\alpha\left(x_i+\frac{k_1}{2}\right)^3\right)\\
k_3 &= \Delta t\left(v_i+\frac{1}{2}l_2\right)\\
l_3 &= \frac{\Delta t}{m}\left(F\left(t_i+\frac{\Delta t}{2}\right)-c\left(v_i+\frac{l_2}{2}\right)-k\left(x_i+\frac{k_2}{2}\right)-\alpha\left(x_i+\frac{k_2}{2}\right)^3\right)\\
k_4 &= \Delta t\left(v_i+\frac{1}{2}l_3\right)\\
l_4 &= \frac{\Delta t}{m}\left(F\left(t_i+\frac{\Delta t}{2}\right)-c\left(v_i+\frac{l_3}{2}\right)-k\left(x_i+\frac{k_3}{2}\right)-\alpha\left(x_i+\frac{k_3}{2}\right)^3\right)
\end{aligned} \qquad \text{(c)}
$$

3.3. Update the displacement and velocity for the next time step:

$$x_{i+1} = x_i + \frac{1}{6}\left(k_1+2k_2+2k_3+k_4\right)$$

$$v_{i+1} = v_i + \frac{1}{6}\left(l_1+2l_2+2l_3+l_4\right)$$

4. Repeat step 3 for each time step until you reach the desired endpoint.

The Runge-Kutta method iteratively approximates the solution to the Duffing oscillator equation by considering the rate of change of displacement and velocity at each time step, providing a numerical solution for $x(t)$ and $v(t)$ over the specified time interval.

B) Duffing Solution

Simple Harmonic:

The temporal evolution and phase-space representation of the solution, denoted as $x(t)$, for the Duffing equation are examined. The Duffing equation is Equation 5.

In Figure 7, consider $\gamma = 0.20$ as a representative amplitude. The remaining parameters are set as follows: $\alpha = -1$, $\beta = +1$, $\delta = 0.3$, and $\omega = 1.2$. The initial conditions for the system are $x(0) = 0.5$ and $\dot{x}(0) = 0$.

The temporal evolution is presented as a time series, where x is plotted as a function of $\frac{t}{T}$, with $T = \frac{2\pi}{\omega}$ representing the period of the oscillation. Additionally, the phase portrait is constructed, depicting the time series plotted in the $x - \dot{x}$ phase plane. Notably, the red dots in the phase portrait correspond to instances when ttt is an integer multiple of the period T.

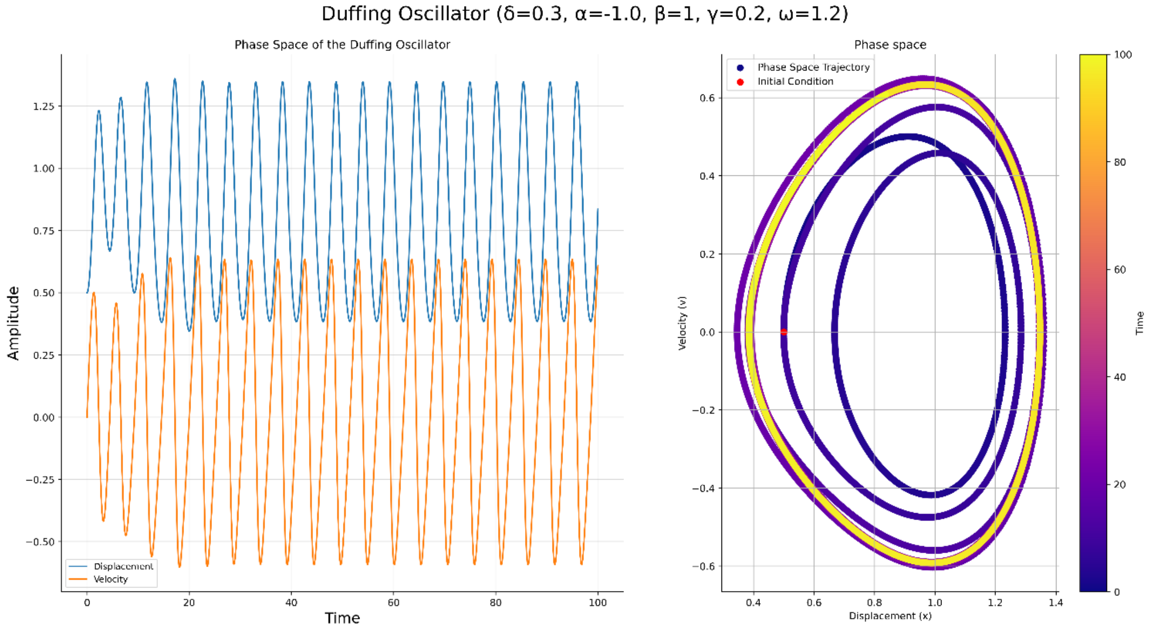

Figure 7. . $\alpha = -1, \beta = +1, \delta = 0.3, \gamma = 0.20$, and $\omega = 1.2$. initial conditions $x(0) = 0.5$ and $\dot{x}(0) = 0$

Quasi-Periodic Behavior

In, Figure 8. consider $\gamma = 0.29$ as a representative amplitude. The remaining parameters are set as follows: $\alpha = -1, \beta = +1, \delta = 0.3$, and $\omega = 1.2$. The initial conditions for the system are $x(0) = 0.5$ and $\dot{x}(0) = 0$.

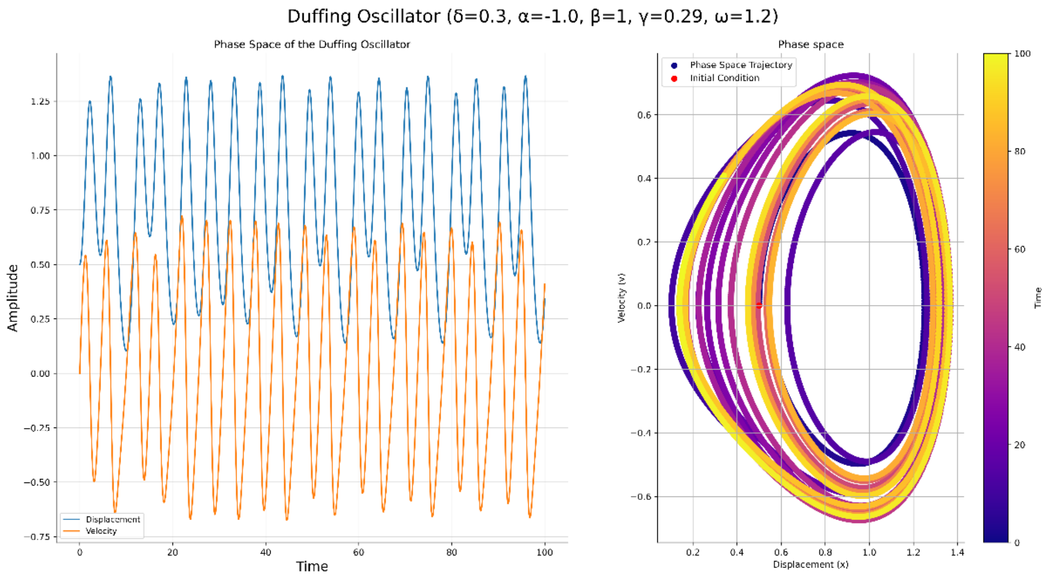

Figure 8. $\alpha = -1, \beta = +1, \delta = 0.3, \gamma = 0.29$, and $\omega = 1.2$. initial conditions $x(0) = 0.5$ and $\dot{x}(0) = 0$

Quasi-Periodic Behavior

In, Figure 8. consider $\gamma = 0.37$ as a representative amplitude. The remaining parameters are set as follows: $\alpha = -1, \beta = +1, \delta = 0.3$, and $\omega = 1.2$. The initial conditions for the system are $x(0) = 0.5$ and $\dot{x}(0) = 0$.

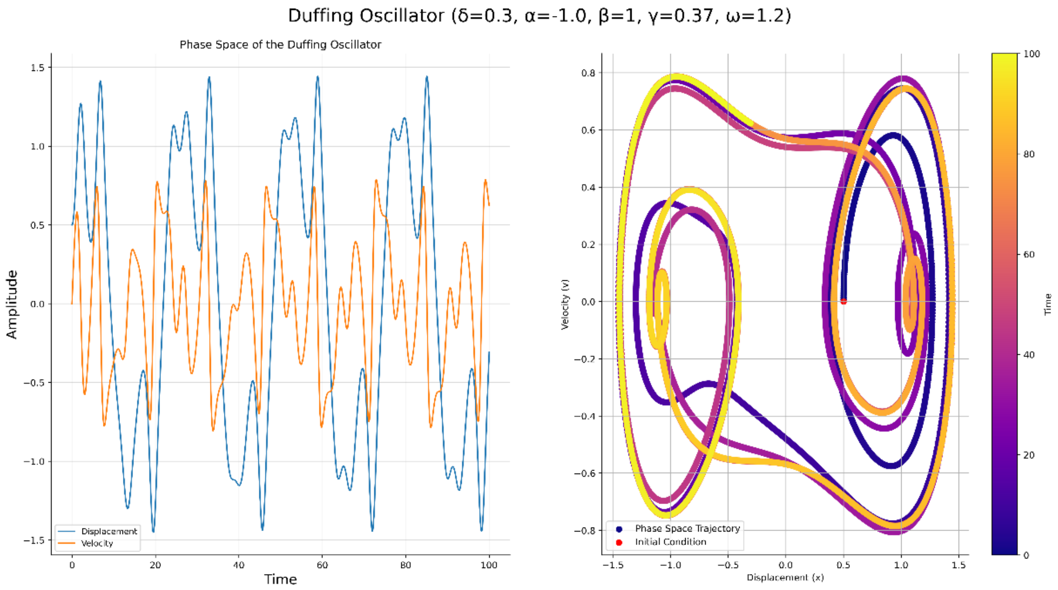

Figure 9. . $\alpha = -1, \beta = +1, \delta = 0.3, \gamma = 0.37$, and $\omega = 1.2$. . initial conditions $x(0) = 0.5$ and $\dot{x}(0) = 0$